Quickly Identifying FFSN Domain and CDN Domain with Little Dataset

Youshuai Zhao, Zhengping Jin

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

15165268607@163.com, zhpjin@bupt.edu.cn

Keywords: little dataset, intrinsic characteristics, quickly identify

Abstract. Fast-flux net has been adopted by more and more attackers to build up bot-net, which consists of many proxy-hosts that are changing fast. With this technology, tracing and locating the C&C (control and command) host is more complex and difficult, so that how to identify whether a domain belongs to a FFSN(fast flux service net) or not is still a very challenging problem. One of these important reasons is that FFSN domains are similar to legitimate CDN(content delivery net) domains.

This paper proposes a new method that uses a little dataset got by active probing, to quickly identify legitimate CDN domains and malicious FFSN domains. Aiming at quickly and effectively identifying domains with little dataset, our proposal is mainly based on FFSN domains' intrinsic characteristics,which are summarized through studying previous researches and data analysis, including features shown in the process of DNS analysis and the process of HTTP visiting. Finally, we use the CART(classification and regression tree) algorithm to train and predict dataset. And the result manifests that the proposed mechanism can be utilized to classify FFSN domains and CDN domains in a few days, which is far less than months taken by previous works, and at a quite high accuracy 90% approximately.

1. Introduction

Bot-net is still one of the biggest risks for net security.With enticing victims to click the malicious hyperlinks, attackers can take control of these hosts,and then C&C(control and command) hosts command these bots to execute vicious services at the same time, such as DDOS(distributed denial of service). The main feature of bot-net which uses FFSN(fast flux service) tech is that users' HTTP requests are redirected to C&C hosts via quick changing bots. Then bot-master can evade detection and blocking, and attackers have the ability to launch large-scale attacks in anytime they want. However, the legitimate CDN(content delivery net) adopts the similar tech, aiming at realizing load balancing and providing good service for users in different regions. Because they are constructed in the similar way, they have similar DNS(domain name system) analysis results, thus to effectively classify them only with DNS records is very difficult.

There have been many researches about FFSN. The widely recognized characteristicsare one domain mapping to several IP addresses which are scatted in different AS (Autonomous systems), and changing fast and having no laws[1-3].Old methods are mainly based on these three characteristics. Researchers collect massive DNS records from DNS servers located in different AS[1], then extract required characteristics, construct features vector for every domain, and finally use one machine learning algorithm to verify their theories. Martinez-Bea.S[4]proposed real-time detection methods with features of bot-net. Hsu C-H[5] also put forward the similar method. These methods are straight and robust[6], but needseveralweeksto collect massive data and to handle the data, which lead to these theories are just theories.

In this paper we propose a new method, aiming at quickly classifying domains' type and not losing accuracy with little dataset. Based onthree characteristics that could greatly determine one domain's type, we used CART(classification and regression tree) algorithm to train and testdataset, then get the final classification accuracy. These three intrinsic characteristics are one domain’s IPs, the IP’s distribution range, and the volatility of visiting time. After studying the processing time of

visiting every selected domain, we find that there exists a big volatility of processing FFSN domains but not in CDN domains. Then we actively collect DNS features of these selected domains with the tool named dig, and get the time of handling a HTTP request through consecutive visiting a domain in a secure environment,and then the volatility could be got.

The rest of this paper is organized as follows. In related work section, we give some researches about FFSN net, and some previous methods to verify FFSN domains. In section 3, we collect the intrinsic characteristics of DNS analyzing records, the nature characteristic of visiting FFSN domain, and do comparison with CDN domains. Then, in methods section, we put forward our method, and givethe detail process of training and testingdataset. In section 5, we accomplish active data collection, preliminarydata analysis, method's verification, and show our final results. Finally, the conclusion is outlined.

2. Related Works

The Honeynet project[7] firstly studied the FFSN net. The researchers explained the underlying operations of bot net bygiving examples of both single and double fast-flux mechanisms. Single fast-fluxmechanisms change the A records of domains rapidly, while double fast-flux techniqueschange both the A records and the NS records of a domain frequently.

There have been many methods toclassify FFSN net and legitimate domains.Roberto Perdisci[3] mentioned passively collected large DNS records from name servers that scattered in different countries and proposed a method based on features of DNS analyzing to cluster and machine learning to determine one domain's type. Martinez-Bea.S[4] proposed real-time detection methods with features of bot-net, which based on features extracted from mass records that were actively collected. Hsu C-H[5] also put forward the similar method. Using these methods could get high accuracy, but need several weeks even several months to collect massive dataand to handle the data. Needing sensors located in many countries and authority problems of some DNS servers make data collecting more difficult. All the above reasons make these methods are theories on the paper. Although the data collection time can be reduced byusing both active and passive monitoring techniques[8], the method still needs mass data and severalminutes along with the help of a data center to determine whether a domain name iscontrolled by a botnet.

Our method is fundamentally different from previous methods. We actively collect data in one selected region in several hours, and do not need help from IDC(Internet Data Center). Based on intrinsic features of FFSN net, we use CART algorithm to train and test our little dataset. As a result, it can classify FFSN and CDN domains efficiently.

3. Intrinsic Characteristics of Fast-flux Net

In this section,we summarize the intrinsic features of DNS analysis, and the nature characteristic of the process of visiting a FFSN domain. Because these characteristics are intrinsic and invariable and cannot be changed by bot-masters, we propose our new methods based on these features.

3.1 Intrinsic Features ofDNS Analysis

Through studying predecessors' researches and works, we conclude that no matter FFSN domain's type is single flux or double flux, it's intrinsic features all include mapping several IPs, and these IPs are located in different network segments.

3.2 Intrinsic Features of Process of Visiting FFSN Domains

Aiming at hiding the origin, the C&C host does not directly handle visitor's HTTP requests. Instead, because mothership has many proxies, it uses one proxy to forward the request to itself, and then use the proxy to forward responses back to the client. We use a packet capturing software to monitor the entire process, which includes the DNS resolution,the process of TCP three-shake, the redirection of the request, and the response's back. Fig 1 shows the entire process.Owing to the definite process of redirection, there will be at least two DNS resolution processes, thus the handling time is longer than that without redirection. And because these proxies'performance ranges from very low to quite high, so there is a big difference between the time of every visiting process.

There will be a big fluctuation of every handling time. Because we are in different regions, we have different net access rate, we may get a long visiting time when we visit a CDN domain. But no matter the status of net, in the same place and during the same period, the fluctuation is invariable, so we select the volatility not the handling time as our next characteristic.

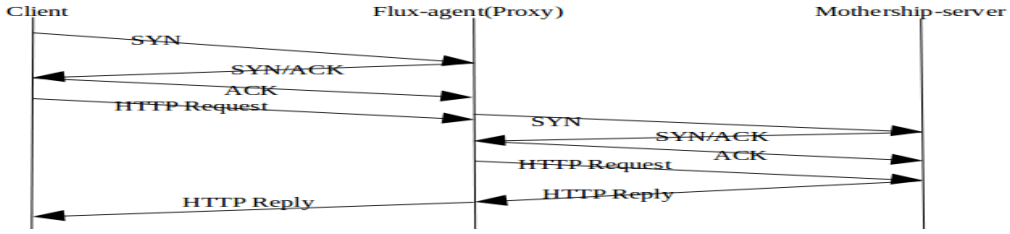

Fig1. The process of visiting a domain which have a redirection

3.3 Comparison with CDN Net Characteristics

CDN,provides services through many nodes which scatted in different locations. Constructing CDN net's target is aiming at quickly reacting with user's HTTP requests with the optimal nodes based on user's geographical positions. So,the DNS resolution of CDN domain will show the similar features of FFSN domain. However, the legitimate corporation's administrator has complete control of all the nodes. So inoneselected region, consecutive DNS resolution results will change in a small scale, and addresses will scatted in less autonomous systems. In the process of handling HTTP requests, there will be no redirection,and because of the average performance of every node, there will be no obvious fluctuation when visiting a CDN domain.

3.4 Summarize

Based on the study in three form parts, we conclude three intrinsic characteristics that could determine one domain's type in a large extent. These characteristics are used in next section, which include IP number, IP16 number, and volatility of visiting time.

4. Methods

4.1 Get the final vectors

Based on the work we have done in the former parts, we finally select the following three important characteristics. And the preliminary verification indeed verify our selection in some extent. We set the characteristics vector related to one domain: $d = (IP\_count, IP16\_count, volatility)$,

- *IP_count* is IPv4 addresses number of *d*
- *IP16_count* is the *IP_count* corresponding type B networks(here we use address's first 16bits)
- *volatility* represents the fluctuation of handling time

The calculation of one domain's volatility: We consecutively visit one domain in the secure environment, and clear the cache in browsers before we visit the domain, then cut the max and min,and get a sequential visiting time $T = (t1, t2, ..., t10)$,

and $M = \left(\sum_{i=0}^{10} t_i \Big/ 10 \right)$,then $volatility = \left(\sum_{i=0}^{10} (t_{i-M})^2 \Big/ 10 \right)$.

Then we get the final vectors , $v_i = (domain, IP\_count, IP16\_count, volatility, class)$

- *class* equals 0 stands for Fast-flux domain, 1 stands for CDN domain.

4.2 Selection of classify method

We select CART algorithm not C4.5 to accomplish the final validation. Because C4.5 algorithm needs scanning and sorting the total data many times in constructing the decision tree, which make this algorithm inefficient in handling big data, however, CART algorithm adopts dividing data space recursively and pruning with validated data, which make it very efficient in classifying mass

data[9]. CART algorithm consecutively executive constructing tree recursively, and then confirm the boundary of every feature to effectively classify domains which are stored in memory, and finally we use these characteristics in predicting phases. And CART algorithm is suited for parallel computation, which is widely used in processing big data. What's more important, CART algorithm has a higher classifying accuracy than C4.5 algorithm. And after doing several comparisons on datasets, we indeed confirm our selection. We can see the advantage of CART in in figure2.

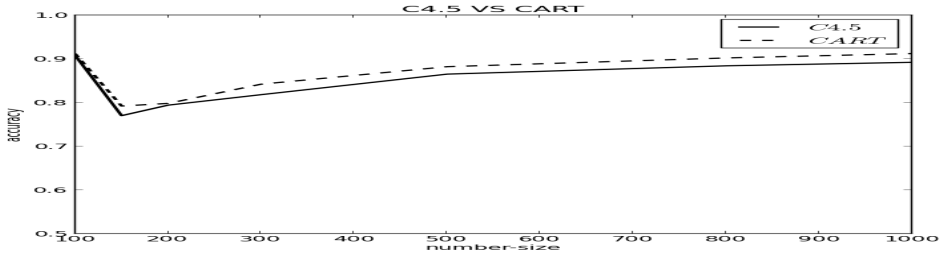

Fig2. Comparison between CART and C4.5

4.3 Train and test

We do the same disposal for every domain, then we stored the required characteristics in MongoDB database. Next we get some domain's records from the database, here we use m CDN domains, n FFSN domains,and then get the dataset $v = [v_1, v_2, ..., v_{m+n}]^T$. Aiming at adding randomness, we executed one random sorting on the dataset. The proportion of training dataset and test dataset is 4:6, and the characteristics is $v_i[1:5]$ label vector is $v_i[5:]$. Then we use CART module of scikit-learn lib written in Python to train the former 40% dataset, then predict a domain's type in the rest 60%dataset. We use the predicted accuracy rate and total processing time to manifest the effectiveness of our result.

5. Data collection and analysis

5.1Active data collection

Firstly, we get malicious domains from malwaredomains.com,ATLAS Global Fast Flux database and FastFlux Tracker in abuse.ch. We collect CDN domains from cdn.h that in github.com/WPO-Foundation. Secondly, we use the tool dig which is prevalent among Unix/Linux users to actively getDNSresolution records of domains listed in the former section, and then summarize every domain's intrinsiccharacteristics.Lastly,we actively visit the domainsthat are outlined in the domain-list,which are executed in virtual machines aiming at not being infected by malwares. Nextwe use the browser's module-Inspect Element(here we use Firefox Browser) to monitor the handling process. We should clear the cache in browsers before we visit the domain, because caches can affect the handling time in some extent. Then, we get the volatility of the visiting time.

5.2 Preliminary Testing and Verification

Here we initially check out the rationality of characteristics that we analyzed and selected.

In the same place, we took five consecutive DNS queries of one Fast-flux domain, divewithshark.nk, and one CDN domainwww.163.com, which were randomly selected, every internal between each query is the TTL. Then we use Python module matplotlib to make a visualization of our results. Fig3 shows the details about divewithshark.nk., and Fig4 shows details about www.163.com.

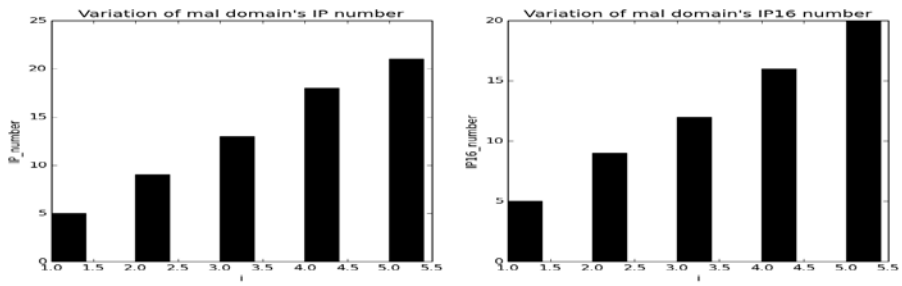

Fig3. Variation of a mal domain's IP number and IP16 number

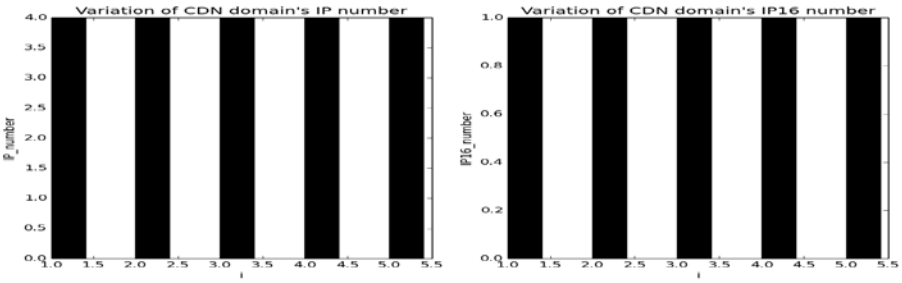

Fig4. Variation of a CDN domain's IP number and IP16 number

We consecutively visit one FFSN domain(here is ather.com) and one CDN domain(here is www.163.com) 12 times, which were randomly selected, and then cut the max and the min, also use Python module matplotlib to make a visualization of the final results. Fig5 shows the variation of visiting time.

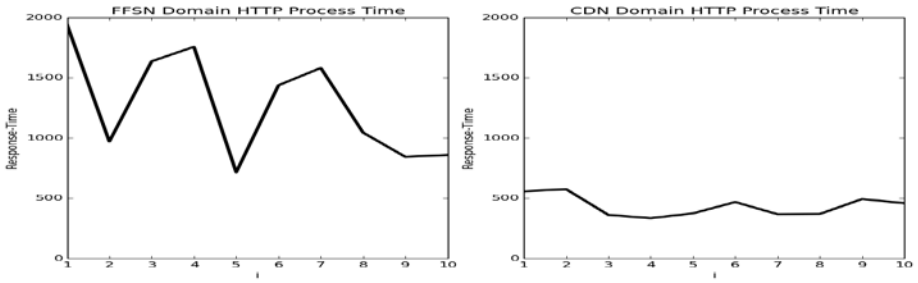

Fig5. Variation of visiting time, left is ather.com, right is www.163.com

5.3 Results analysis

Here, we perform a cross validation to estimate the accuracy of our method. We use CART module in python's lib scikit-learn to accomplish the final training and testing process in our method. Because there are few flux domains in hundreds gigabit data, for the max accuracy, we select some domains whose type is known. Here, we set $m = 72$,and $n = 28$. Considered that every classification process has randomness in some extent, so we here execute 12 times, then discard the maximum result and the minimum result, the final average classification accuracy is 89.1%. Fig6 shows the variation of every result. The results show that we can use little dataset to identify FFSN and CDN domains with quite high accuracy and we can accomplish all procedures in about four hours.

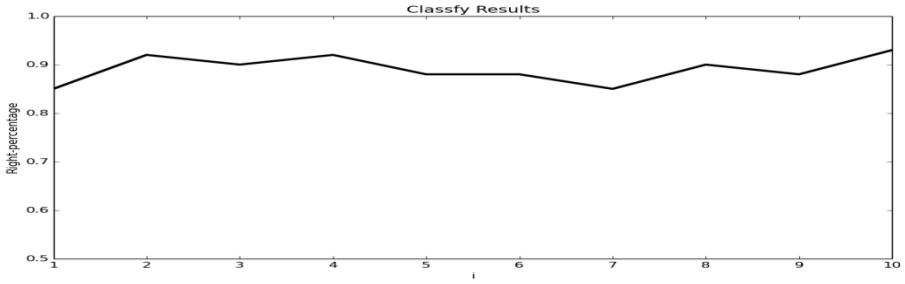

Fig6. Variation of classification results

6. Conclusion

In this paper, we proposed a new method to quickly and effectively classify CDN domains and FFSN domains. Based onthree intrinsic characteristics that could greatly determine one domain's type, we use CARTalgorithm to train and test the little dataset that are actively collected, then get the final classification accuracy.The method not only evades the weakness that taking vast huge manpower and long time, but also uses these two widelyrecognized characteristics,IP_count and IP16_count,what'smore important,we also uses a new intrinsic characteristic found in the process of visiting one selected domain which can differentiate domains obviously. We can use the proposed method with little dataset to identify one domain's type at an average percentage 90% accuracy, and we can accomplish the overall process in several hours.

Acknowledgements

This work is supported by NSFC (Grant Nos. 61300181, 61502044), the Fundamental Research Funds for the Central Universities (Grant No. 2015RC23).

Reference

[1] H.wang, C.Mao, K.Wu, H.Lee. *"Real-time fast-flux identification via localized spatial geolocation detection"*. In: Proc. Computer Software and Applications Conference(COMPSAC),2012, p244-252.

[2] Celik Z.B, Oktug S. *"Detection of Fast-Flux Networks using various DNS feature sets"*. In: Computers and Communications (ISCC), 2013 IEEE Symposium, p868-873.

[3] Roberto Perdisci, Igino Corona, Giorgio Giacinto. *"Early detection of malicious flux networks via large-scale passive DNS analysis"*. In: Dependable and Secure Computing, IEEE Transactions, 2012, Vol. 9 Issue 5, p714-726.

[4] Martinez-Bea S, Castillo-Perez S, Grcia-Alfaro J. *"Real-time malicious fast-flux detection using DNS and bot related features"*. In: 2013 Eleventh annual international conference on privacy, security and trust (PST), 2013, p369-372.

[5] Basheer N.AI-Duwairi, Ahmad T. AI-Hammouri. *"Fast Flux Watch: A mechanism for online detection of fast flux networks"*. In: Joural of Advanced Research, May2014, p473-479.

[6] Hsu C-H, Huang C-Y, Chen K-T. *"Fast-flux bot detection in real time"*. In: Proceedings of the 13th international conference on Recent advances in intrusion detection, Oct2010, p464-483.

[7] *The Honeynet project, know your enemy: fast-flux service networks [Internet]*, 2007 [cited 27.08.13], <http://www.honeynet.org/book/export/html/130>.

[8] CaglayanA,ToothakerM, DrapeauD,BurkeD, EatonG. "Real-time detection of fast flux service networks". In: Proceedings of the Cybersecurity Applications & Technology Conference for Homeland Security, 2009, Vol.0 p285-292.

[9] Yong-Gyu Jung, Jun Heo, Kyu-Ho Kim. "Comparative study on changes in performance of C4.5 and CART algorithms". In: Journal of Next Generation Information Technology, Aug2013, Vol. 4 Issue 6, p28.